University of Glasgow

Internet Protocols Laboratory

Real-time and Interactive Applications

Networked Systems (H)

Lecture 7

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow | This work is licensed under the Creative Commons Attribution-NoDerivatives 4.0 International License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Lecture Outline

- Real-time applications
  - What is real-time traffic?
  - Requirements and constraints
  - Quality-of-service and quality-of-experience
- Interactive applications
  - Conferencing application architecture
  - Signalling, session description, and real-time media
- Streaming applications
  - HTTP adaptive streaming

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

University of Glasgow

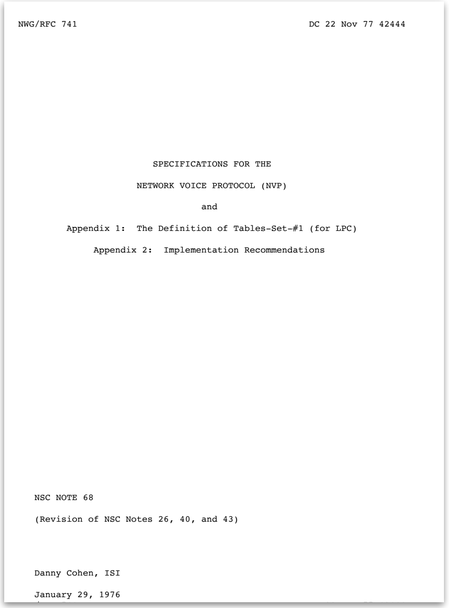
Real-time Applications

- What is real-time traffic?
- Requirements and constraints
- Quality of service vs. user experience

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Real-time Traffic on the Internet

- Long history of real-time traffic over the Internet:
  - Telephony and voice-over-IP (VoIP), Internet radio, video conferencing, streaming video and TV, gaming, sensors and industrial IoT
  - 1970s – Initial experiments, Network Voice Protocol
  - 1990s – Core standards defined, SIP, SDP, RTP, RTSP, H.323
  - 2010s – HTTP adaptive streaming, MPEG DASH
- The design of the network and its transport protocols has evolved to support real-time media



NWG/RFC 741 DC 22 Nov 77 42444

SPECIFICATIONS FOR THE
NETWORK VOICE PROTOCOL (NVP)
and
Appendix 1: The Definition of Tables-Set-#1 (for LPC)
Appendix 2: Implementation Recommendations

NSC NOTE 68
(Revision of NSC Notes 26, 40, and 43)

Danny Cohen, ISI
January 29, 1976

https://datatracker.ietf.org/doc/rfc741/

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

What is Real-time Traffic?

- Defining characteristic: **deadlines**
  - The system fails if data is not delivered by a certain time
    - Railway signalling must deliver data before train arrives – **hard real-time**
    - Streaming video should deliver new frames regularly else playback stutters – **soft real-time**
    - What is the probability a deadline is missed? What are the consequences? Engineer accordingly – **no system is 100% reliable**
  - Deadlines can be absolute or relative
    - Data must be delivered before a certain time – **absolute deadline**
    - Data must be delivered periodically, within some time after the previous – **relative deadline**
- Real-time is not necessarily high performance
  - It requires **predictable timing**, not necessarily high bandwidth or low latency

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Requirements for Streaming Applications

- Streaming on demand video has no absolute deadlines, needs regular data to prevent stalls once started
  - May be acceptable if a takes a few seconds to start the video
  - Playback should be smooth once it starts
- Streaming **live** video (e.g., live sports) may have absolute deadlines
- Bit rate depends on desired quality
  - Higher quality is better, up to limits of display
  - Predictable, but lower, quality often preferred to more variable, but on average higher, quality
  - For a given bit rate, trade-off between frame rate and frame quality

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Requirements for Interactive Applications

- Requirements for interactive applications determined by task and human perception
  - Phone call or video conference
    - One-way mouth-to-ear delay ~150ms maximum for telephony
    - For lip-sync, want audio <15ms ahead, <45ms behind, video
  - Lecture style, mostly unidirectional with occasional questions, tolerates much higher latency
  - Distributed music performance
    - One-way latency ≪50ms desirable
    - Speed of sound: ~15ms to go from one side of a large orchestra to the other



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.114 (05/2003)

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

International telephone connections and circuits – General Recommendations on the transmission quality for an entire international telephone connection

One-way transmission time

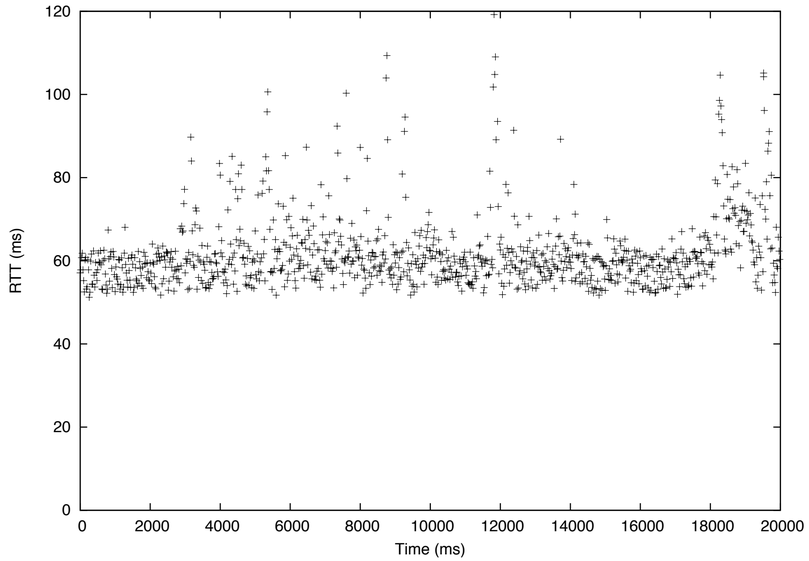
ITU-T Recommendation G.114

ITU Recommendation G.114

ATSC IS-191 “Relative Timing of Sound and Vision for Broadcast Operations”, 2003

CC BY ND

Real-time Traffic Must Be Loss Tolerant



- The Internet is a best effort packet network
  - Timing is not guaranteed – depends on the amount of queueing on the path and on the path taken
  - Packets can be lost – retransmissions take time, and may not arrive before the deadline
  - In some networks these effects can be significant – with careful engineering they can be insignificant
- Real-time applications need to be loss tolerant

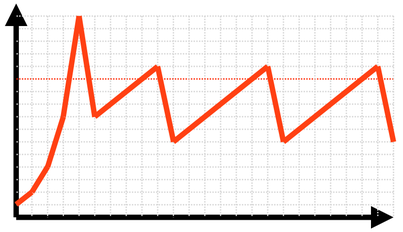


Data transport can be timely or reliable: not possible to be both simultaneously

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Real-time Traffic Has Limited Elasticity

- TCP congestion control adapts transmission speed to match capacity
- Many transfers are **elastic** – faster is better, but it doesn't matter what rate the congestion control selects
  - The application can adapt to any chosen rate
- Real-time traffic is **inelastic**
  - Media has a minimum rate, below which it cannot be used
  - Media has a maximum rate and cannot consume more



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Quality of Service (QoS) Guarantees

- The network can prioritise real-time traffic based on DSCP field in IP header
- The network operator can also reserve capacity for real-time traffic
  - e.g., 5G wireless “network slicing” to allocate capacity to particular applications
- Requires signalling, authorisation, and accounting
  - Need to tell the network what resources the real-time traffic requires
  - Need to demonstrate that sender is allowed to reserve these resources and can pay for the reserved capacity; reservations prevent other users from accessing the network, so have some cost

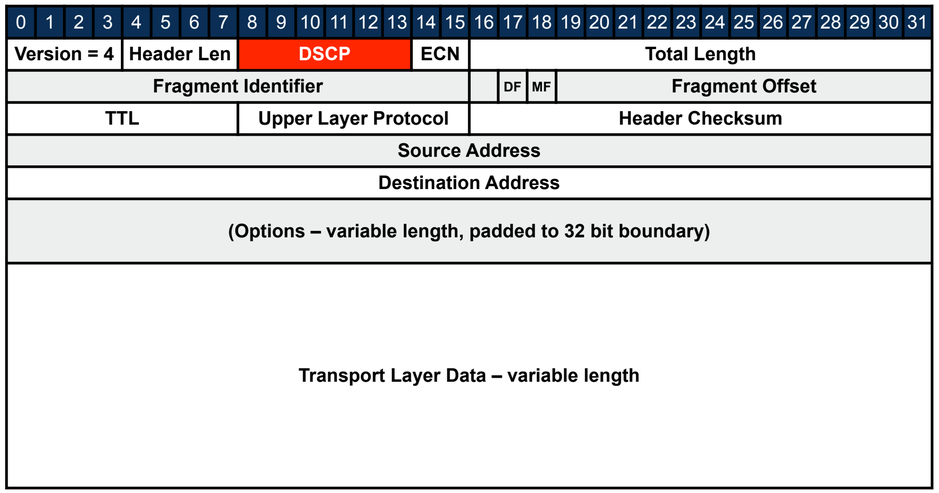


Diagram is IPv4; IPv6 works the same

- Does this improve performance?
- What are the anti-trust risks?
- What are the human rights risks?

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Quality of Experience (QoE)

- What ultimately matters is subjective quality of experience
  - Does the application meet user needs?
  - Does the application allow users to communicate effectively?
  - Does the application provide compelling entertainment?
- QoE is not a one-dimensional metric
  - What aspect of user experience are you evaluating?
    - e.g., “Does it sound good” is not the same as “Can you understand it?”
  - How does your metric relate to the task being performed?
  - Some aspects of user experience can be estimated from technical metrics
  - Some aspects are subjective and task dependent – need user trials

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

University of Glasgow

Interactive Applications

- Structure of video conferencing systems
- Protocols for video conferencing
- Media transport

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Interactive Conferencing Applications

- What are interactive applications?
  - Telephony
  - Voice-over-IP (VoIP)
  - Video conferencing
- Long history of research and standardisation:
  - Network Voice Protocol (NVP)
  - Modern standards development from mid-1990s
  - Browser-based conferencing from mid-2010s

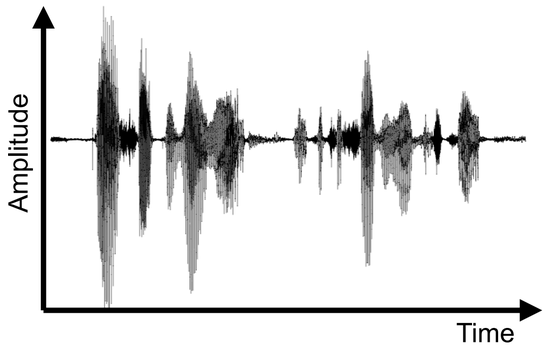


Multi-party video conferencing, 1998

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Constraints

- Interactive applications have strict timing constraints
  - One-way mouth-to-ear delay <150ms, lip-sync audio and video
  - Audio encoded in 20ms frames
    - Speech codecs capture, encode, and transmit one frame of speech every 20ms; data rate tens of kilobits per second
    - Music has similar timing requires, but hundreds of kbps
  - Video frame rate and resolution highly variable
    - High definition video using H.264 is around 2-4Mbps; 25-60fps commonly used
    - I-frames → tens of packets; P-frames → single or few packets



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Reliability

- Speech data is highly loss tolerant
  - Loss concealment can hide 10-20% random packet loss without noticeable loss in quality
  - Burst losses are less well concealed
- Video packet loss hard to conceal
  - No scene changes to reset decoder state to known good value
  - Retransmissions possible in some cases; forward error correction more typical

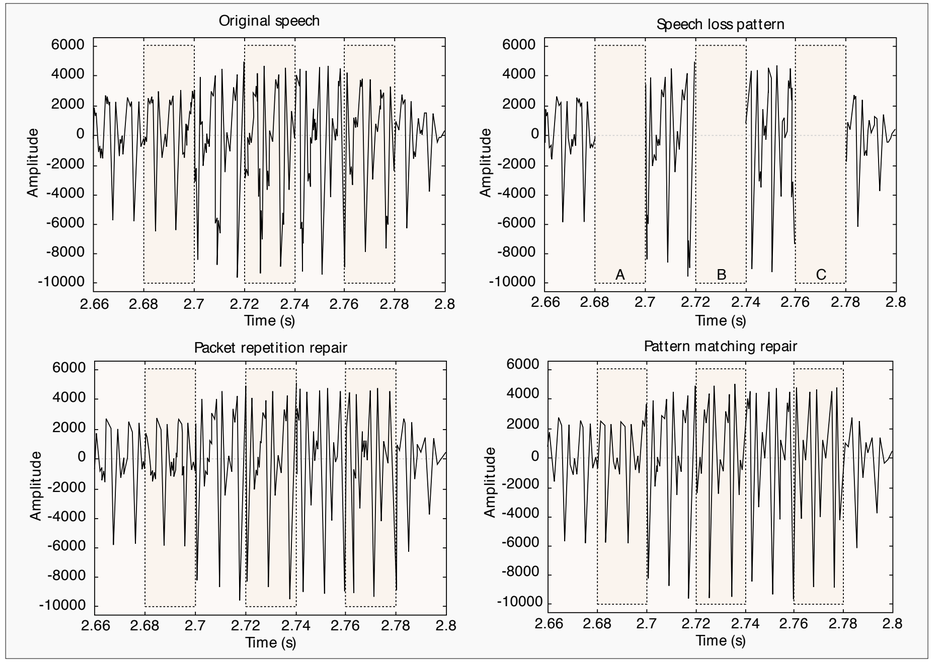
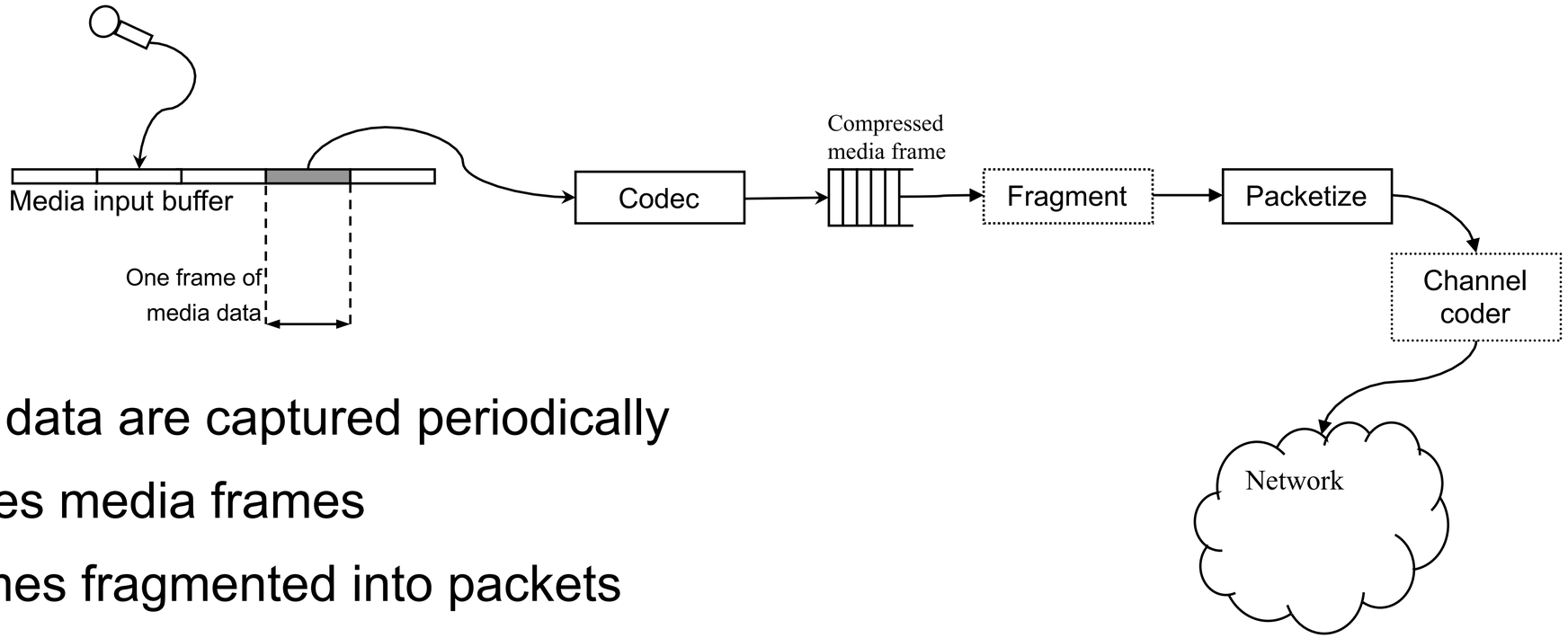


Figure 9. *(a) Sample error concealment techniques: original audio signal; (b) sample error concealment techniques: the loss pattern; (c) sample error concealment techniques; packet repetition; (d) sample error concealment techniques: one sided waveform substitution.*

C. S. Perkins, O. Hodson, and V. Hardman, A Survey of Packet Loss Recovery Techniques for Streaming Media, IEEE Network Magazine, September 1998. DOI:10.1109/65.730750

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transmission Path

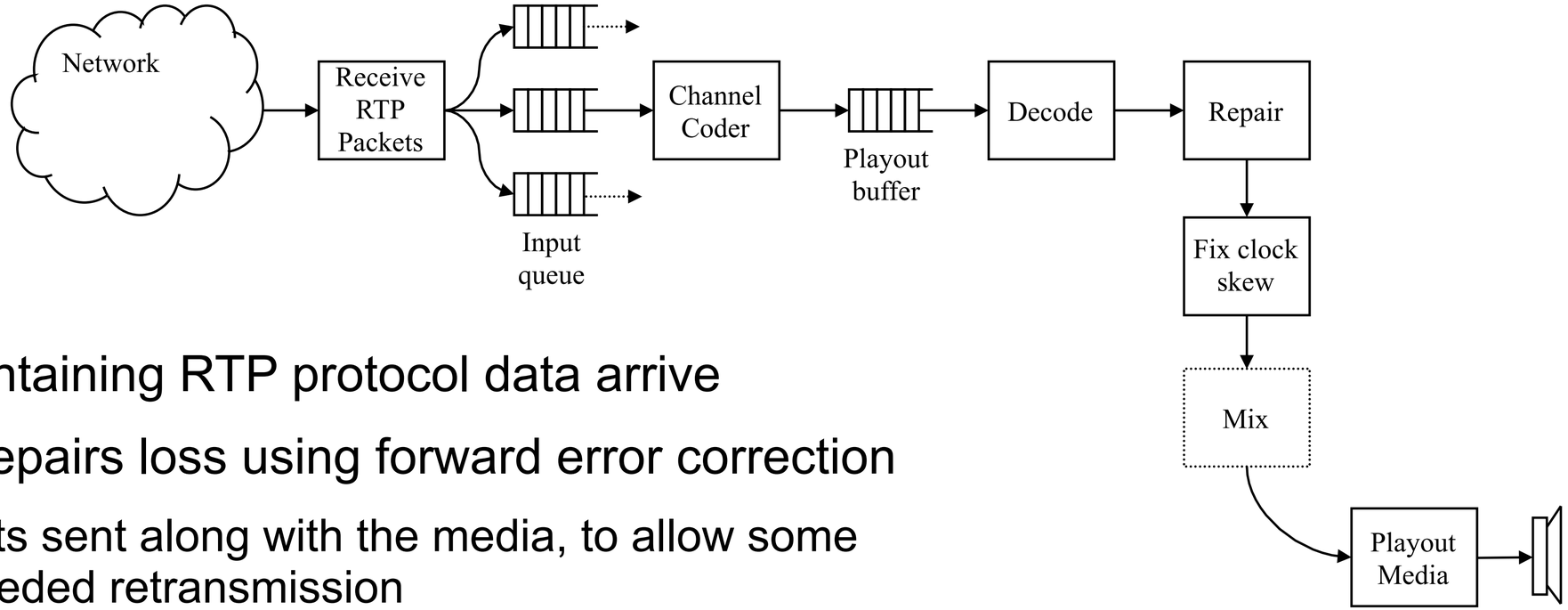


- Frames of media data are captured periodically
- Codec compresses media frames
- Compressed frames fragmented into packets
  - Transmitted using RTP inside UDP packets
  - RTP protocol adds timing and sequencing, source identification, payload identification
- Transmitted over the network

Source: C. Perkins. "RTP: Audio and Video for the Internet", Addison-Wesley, 2003

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Reception Path

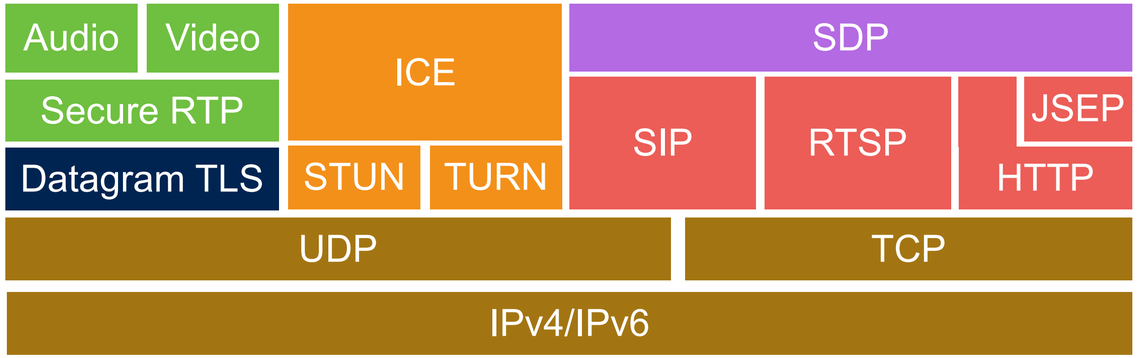


- UDP packets containing RTP protocol data arrive
- Channel coder repairs loss using forward error correction
  - Additional packets sent along with the media, to allow some repair without needed retransmission
- Playout buffer used to reconstruct order, smooth timing
- Media is decompressed, packet loss concealed, and clock skew corrected
- Recovered media is played-out to user

Source: C. Perkins. "RTP: Audio and Video for the Internet", Addison-Wesley, 2003

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

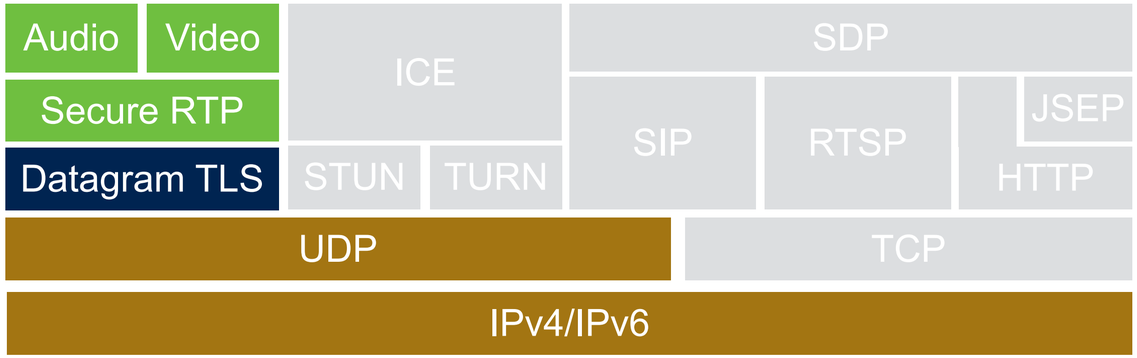
Standards for Interactive Conferencing



- Media
- Path discovery and NAT traversal
- Session descriptions
- Signalling protocols
  - Control of streaming media
  - Control of interactive conferencing
  - Control of web-based interactive media

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

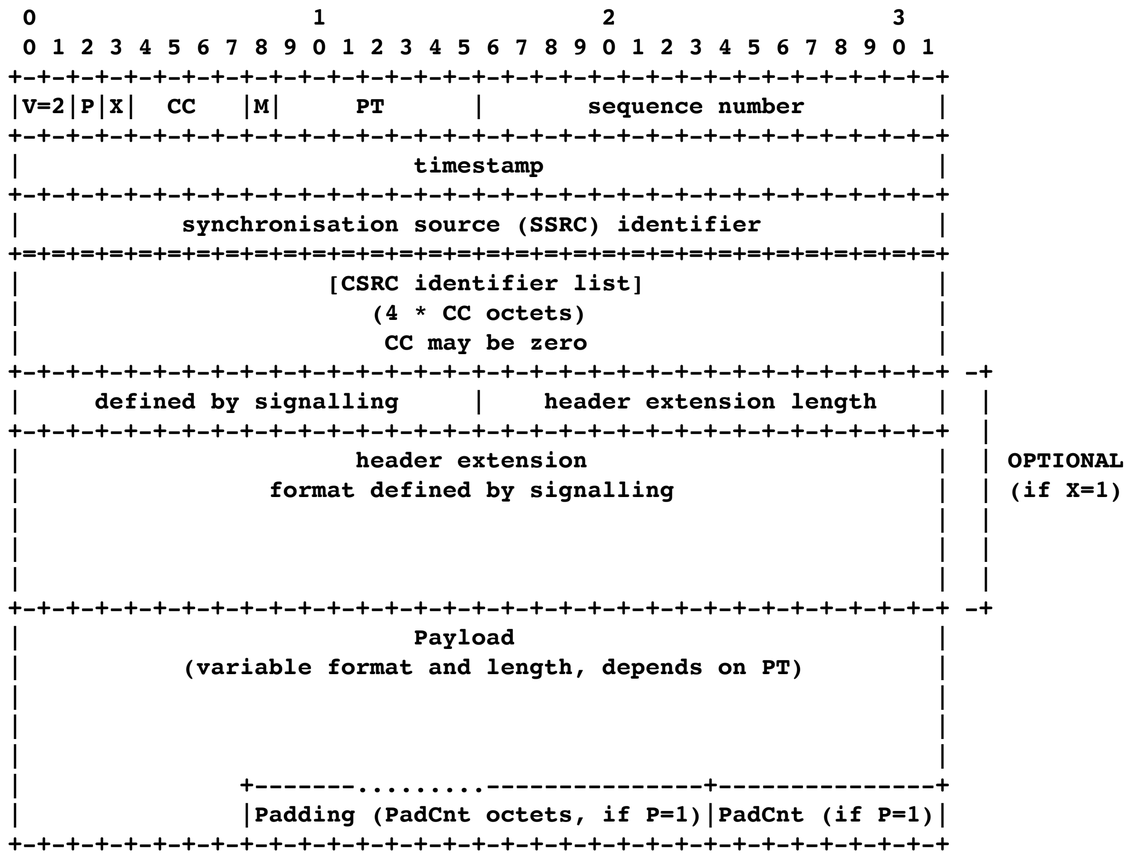
Media Transport



- Audio and video data compressed and sent inside Secure **RTP** packets
- Security is provided by Datagram TLS
  - TLS 1.3 handshake packets exchanged in UDP to agree encryption keys
  - Encryption keys are used to encrypt the RTP packets
- RTP packets are sent inside UDP packets – prefer timely to reliable

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transport: RTP

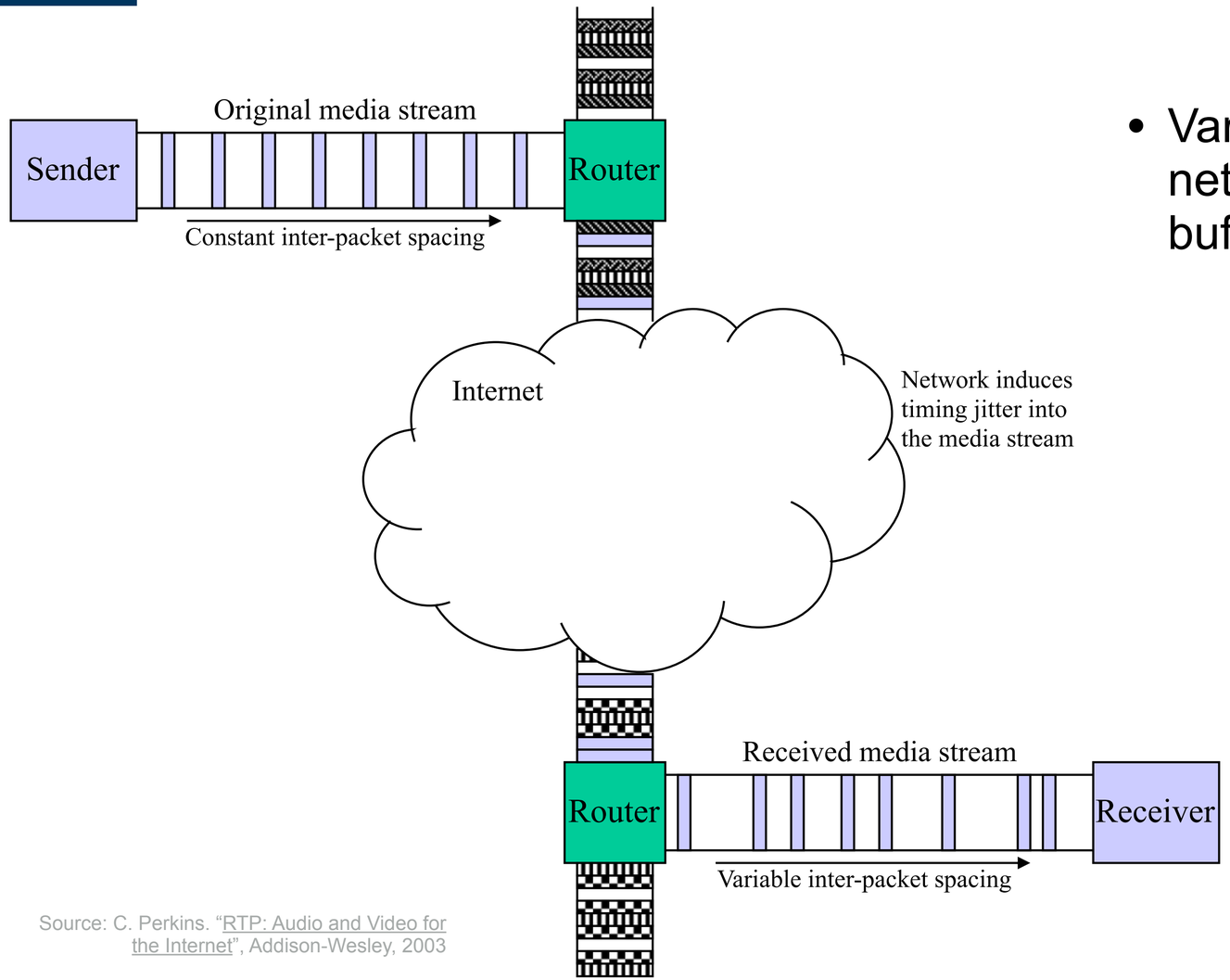


https://datatracker.ietf.org/doc/rfc3550/

- RTP data packets carried within UDP
- Header information carries:
  - Sequence number and timestamp
    - Allows receiver to reconstruct ordering and timing
  - Source identifiers
    - Who sent this packet – needed for multiparty calls
  - Payload format identifier
    - Does the packet contain audio or video?
    - What compression algorithm is used?

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transport: Timing Recovery (1/3)

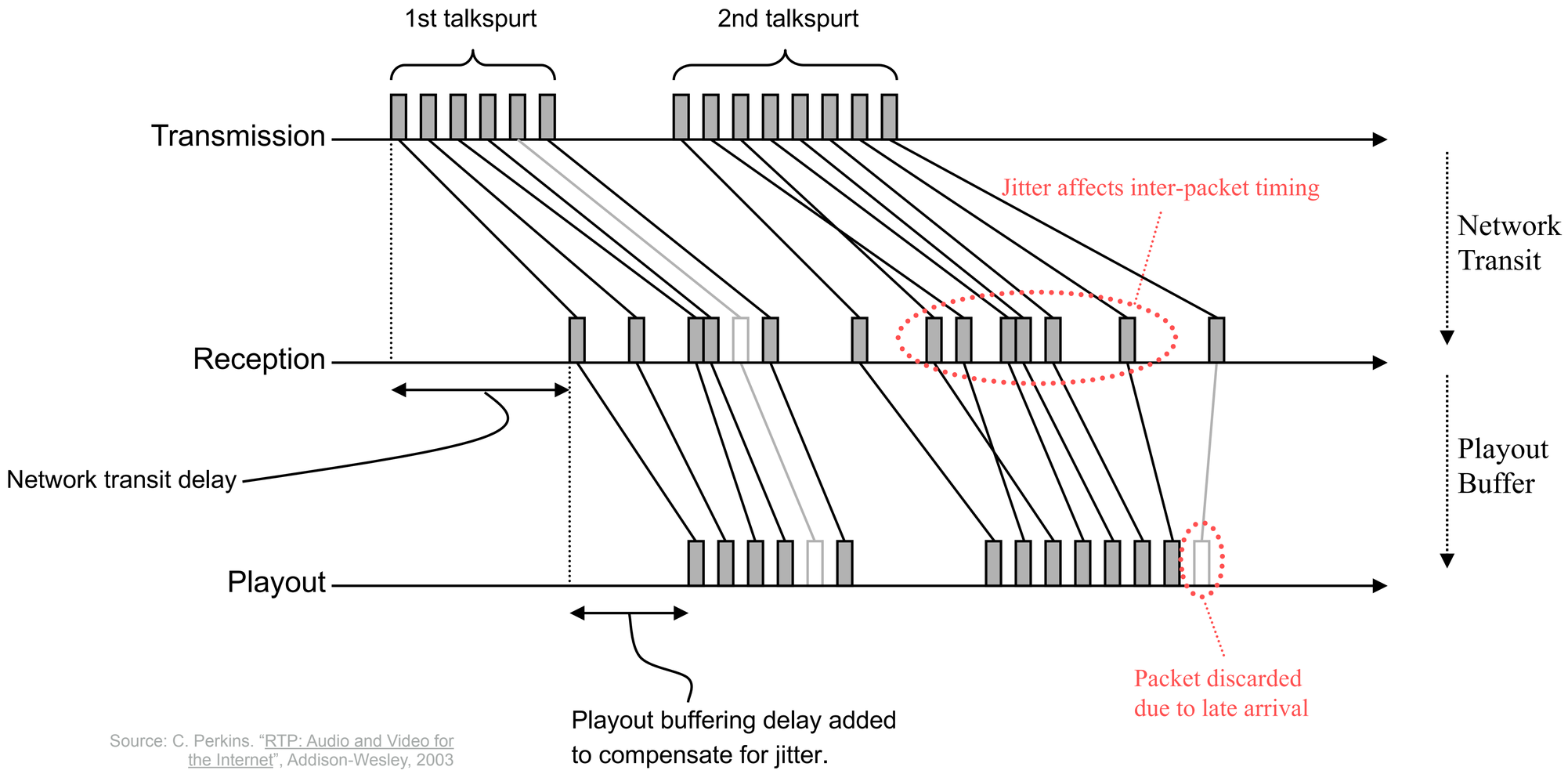


Source: C. Perkins. "RTP: Audio and Video for the Internet", Addison-Wesley, 2003

- Variable queueing delays in the network disrupt timing – receiver buffers to smooth out variation

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transport: Timing Recovery (2/3)

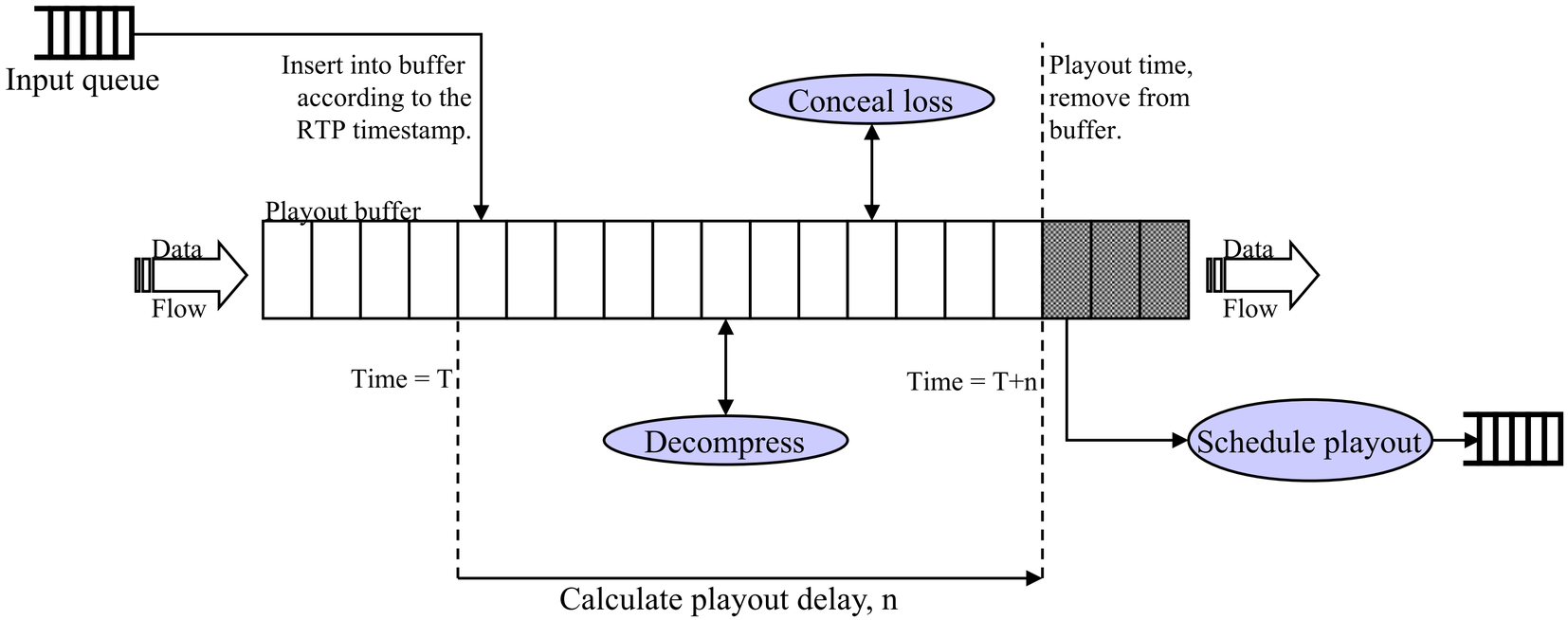


Source: C. Perkins. "RTP: Audio and Video for the Internet", Addison-Wesley, 2003

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transport: Timing Recovery (3/3)

- If packets played out immediately on arrival, variation in timing leads to gaps
- Delay playout by more than typical variation in inter-arrival time, to allow smooth play back



Source: C. Perkins. "RTP: Audio and Video for the Internet", Addison-Wesley, 2003

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transport: Application Level Framing

- Packet loss is possible, so receivers must make best use of packets that do arrive
- RTP **payload formats** define how compressed audio/visual data is formatted into RTP packets
  - Goal: **each packet should be independently usable**
  - If a packet arrives, it should be possible to decode all the data it contains – not always possible, but desirable
  - Naïve packetisation can lead to inter-packet dependencies where a packet arrives but can't be decoded because some previous packet, on which it depends, was lost



Architectural Considerations for a New Generation of Protocols

David D. Clark and David L. Tennenhouse
Laboratory for Computer Science, M. I. T.

Abstract

The current generation of protocol architectures, such as TCP/IP or the ISO suite, seem successful at meeting the demands of todays networks. However, a number of new requirements have been proposed for the networks of tomorrow, and some innovation in protocol structuring may be necessary. In this paper, we review some key requirements for tomorrow's networks, and propose some architectural principles to structure a new generation of protocols. In particular, this paper identifies two new design principles, Application Level Framing and Integrated Layer Processing. Additionally, it identifies the presentation layer as a key aspect of overall protocol performance.

1 Introduction

Historically, there has been a sharp distinction between the network architectures used in universal access environments, such as the telephone network, and the architectures developed within the computer communications community. We believe that both communities would benefit from a unified architecture that addresses new communication requirements. We have identified three key requirements that will stress the existing protocol architectures. Future networks will have considerably greater capacity, will be based on a wider selection of technologies, and will support a broader range of services.

The principle design goal for many research projects, and some commercial products, is gigabit operation, both in aggregate over trunks and to individual end points. As networks proceed to higher speeds, there is some concern that existing protocols will represent a bottleneck, and various alternatives have been proposed, such as outboard protocol processors [1, 11] and new protocol designs [2, 14, 12]. Since there is no clear consensus on the real source of protocol overhead, it is hard to validate these various solutions.

Another requirement is that the protocols of tomorrow operate over the range of coming network technology. An obvious example of emerging technology is Broadband ISDN [19, 6] which is based on Asynchronous Transfer Mode (ATM). This network technology provides a switching mode somewhat different from classic packet switching in that the data to be transferred is divided up into small (currently 48 byte) fixed size elements for transmission. A more futuristic network technology is wavelength division over fibre.

Finally, there is the requirement for service integration. In the future, a single end system will be expected to support applications that orchestrate a wide range of media (voice, video, graphics, text) and access patterns (interactive, bulk transfer, real-time rendering). This cross-product of media and access will generate new traffic patterns with performance considerations, such as delay and jitter tolerance, that are not addressed by present architectures.

In summary, future networks must exhibit a significant degree of flexibility and be based on an architecture that admits a wide range of application demands and implementation strategies. In the remainder of this paper, we identify some key principles for the elaboration of such an architecture.

2 Structuring Principles for Protocol Architectures

Most discussions of protocol design refer to the ISO reference model [9], or some other model that uses layering to decompose the protocol into functional modules. There is, however, a peril in using only a layered model to structure protocols into components, which is that layering may not be the most effective modularity for implementation.

It is important to distinguish between the architecture of a protocol suite and the engineering of a specific end system or relay node. The architecture specifies the decomposition into functional modules, the semantics of the individual modules and, the syntax used to effect the protocol. There should be no *a priori* requirement that the decomposition of the engineering design of a given system correspond to the architectural decomposition of the protocol. In the case of layered architectures, practical experience [4, 17] provides strong support for alternative engineering designs. Unfortunately, the structure of the protocol architecture may unnec-

*This research was supported by the Defense Advanced Research Projects Agency, monitored by the National Aeronautics and Space Administration under contract No. NAG2-581, and by the National Science Foundation under grant NCR-8414187.

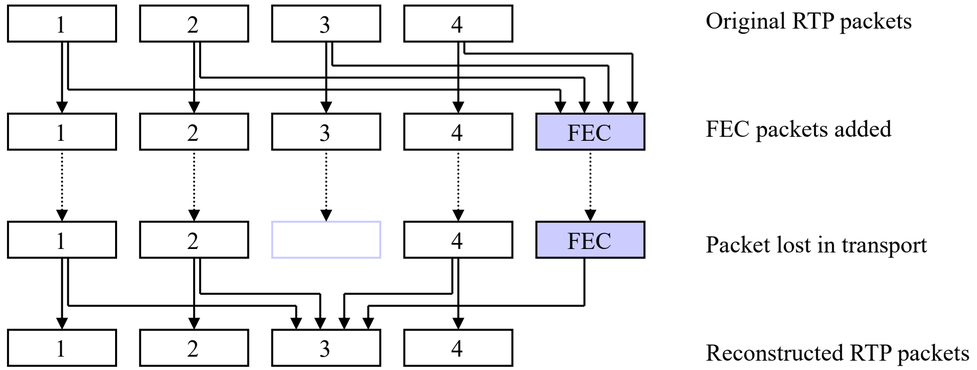
Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1990 ACM 089791-405-8/90/0009/0200...$1.50

D. Clark and D. Tennenhouse, "Architectural considerations for a new generation of protocols", ACM SIGCOMM Conference, September 1990. https://dx.doi.org/10.1145/99508.99553/

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Media Transport: FEC vs. Retransmission

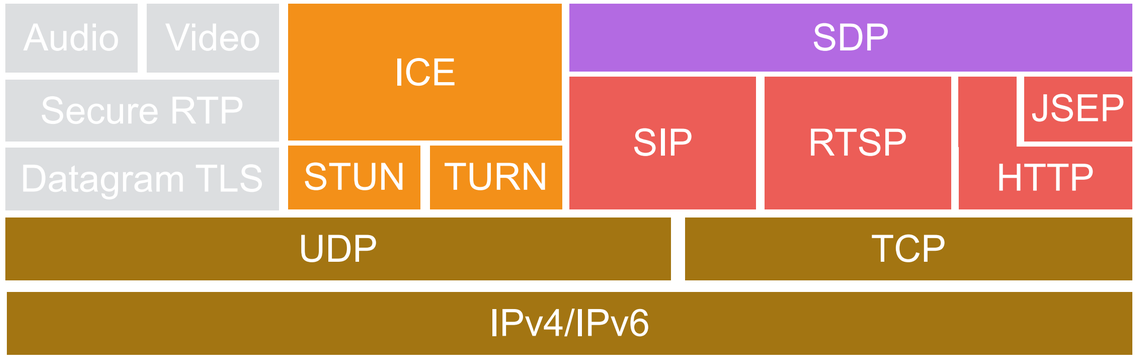
- Retransmission possible, but often takes too long – packet should have been played out before retransmission arrives
- Forward error correction (FEC) often used instead



- Additional FEC packets are sent along with the original data
  - Contain error correcting codes
  - e.g., the Exclusive-OR (XOR) of the original packets – many different FEC schemes
- If some original packets are lost but the FEC packets arrive, original data can be reconstructed

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Signalling



- Signalling protocols find the called party, negotiate compression algorithms, and establish the call

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Signalling and Session Descriptions

- Media transport flows peer-to-peer for low latency
- A **signalling protocol** is needed to find the peer and establish the paths
- The control protocol needs to describe the communication session expected:
  - Media transport connections required
  - Media formats and compression algorithms
  - IP addresses and ports to use
  - Originator and purpose of session
  - Options and parameters

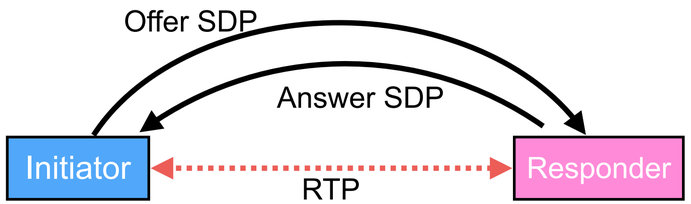
```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 99
a=rtpmap:99 h263-1998/90000
```

M. Handley, V. Jacobson, and C. Perkins, “SDP: Session Description Protocol”, IETF, RFC 4566, July 2006. https://datatracker.ietf.org/doc/rfc4566/

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Session Descriptions: SDP Offer/Answer

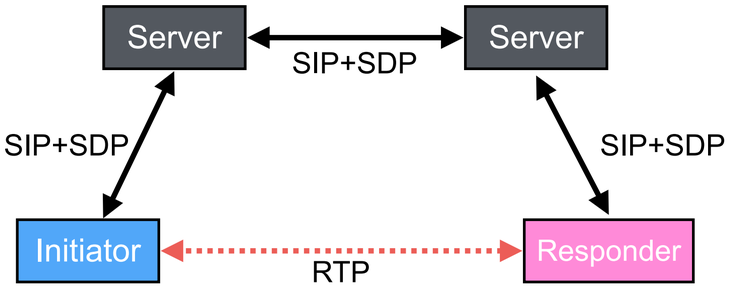
- Interactive sessions require negotiation
  - An **offer** to communicate: lists codecs, options and addressing details, identity of caller
  - The **answer** subsets codecs and options to those mutually acceptable, supplies addressing details, and confirms willingness to communicate
  - NAT traversal algorithms used to find candidate addresses
- SDP used as the negotiation format
  - SDP **was not** designed to express options and alternatives
  - Insufficient structure in syntax, semantic overloading
  - Complex → but complexity not initially visible; now too entrenched for alternatives to take off



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Control of Interactive Conferencing: SIP

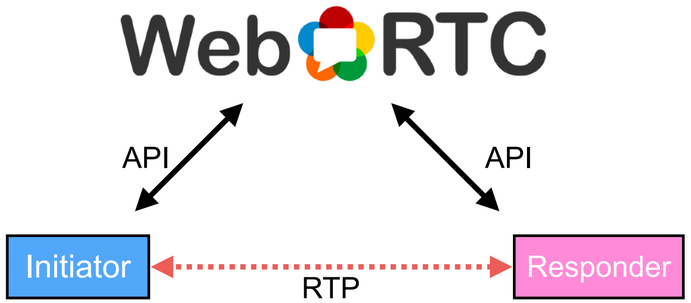
- The signalling protocol sets up the call
  - **Session Initiation Protocol** (SIP) for telephony and video conferencing
  - Conferencing servers with public IP addresses handle calls for a domain
  - SIP messages sent from initiator to responder via servers
    - Determine location of responder, discover NAT bindings, negotiate parameters for the call
    - An offer/answer exchange using SDP to describe the session being created
  - Alert the responding user – make phone ring! – and agree to setup a call
    - NAT binding discovery and connection probing takes place while alerting
  - Media then flows over direct peer-to-peer path



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Browser-based Conferencing: WebRTC

- **WebRTC** is an alternative signalling protocol
  - Implemented in web browsers
  - Exposes standard JavaScript API
  - Signalling messages delivered via HTTP to web server controlling the call
    - Offer/answer exchanges using SDP
  - Media transport uses RTP – same as SIP – with peer-to-peer data channel
  - Designed to integrate video conferencing into web browsers and web applications



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

University of Glasgow

Streaming Video

- HTTP Adaptive Streaming

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Streaming Video Applications

- RTP **should** be usable for streaming video
  - Real-time Streaming Protocol (RTSP) developed by Real Networks for Internet TV in late 1990s
  - Uses unidirectional RTP for video delivery
  - Very low latency, loss tolerant

- **Most** video-on-demand, TV, and movies are delivered instead over HTTPS
  - Performance is significantly worse, very high latency, but startup latency is less critical
  - Integrates better with content distribution networks



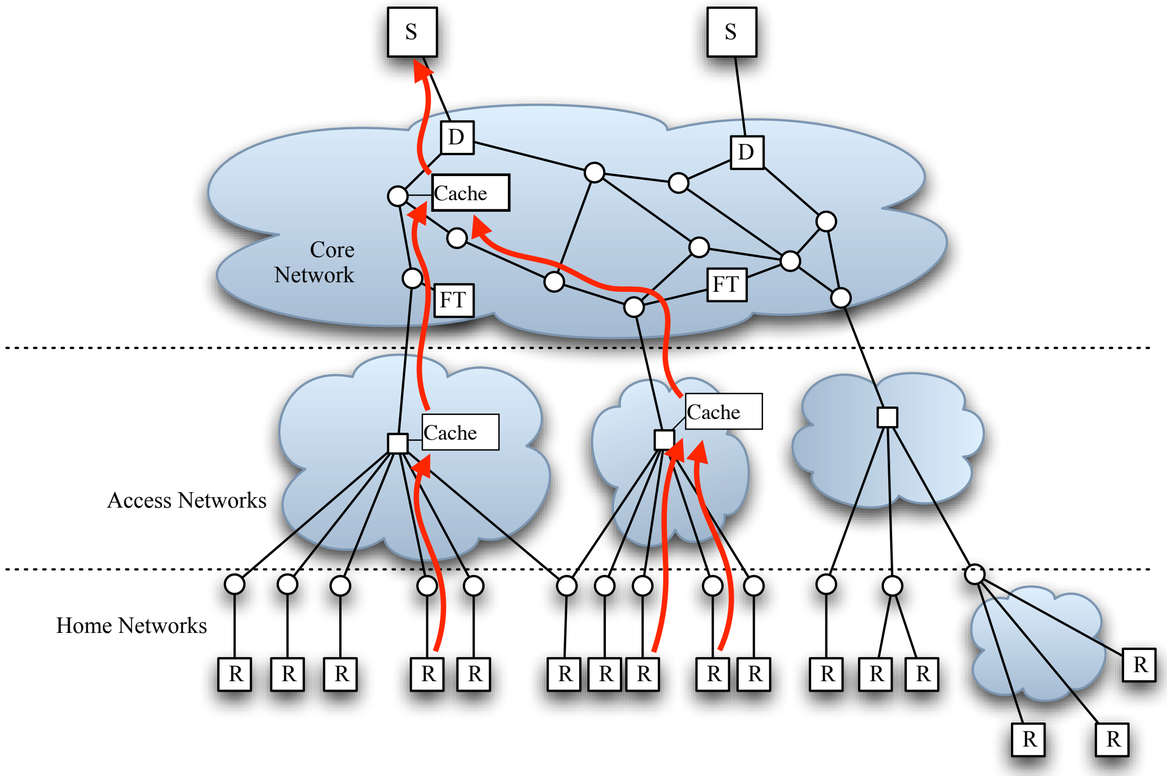






Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

HTTP Content Distribution Networks



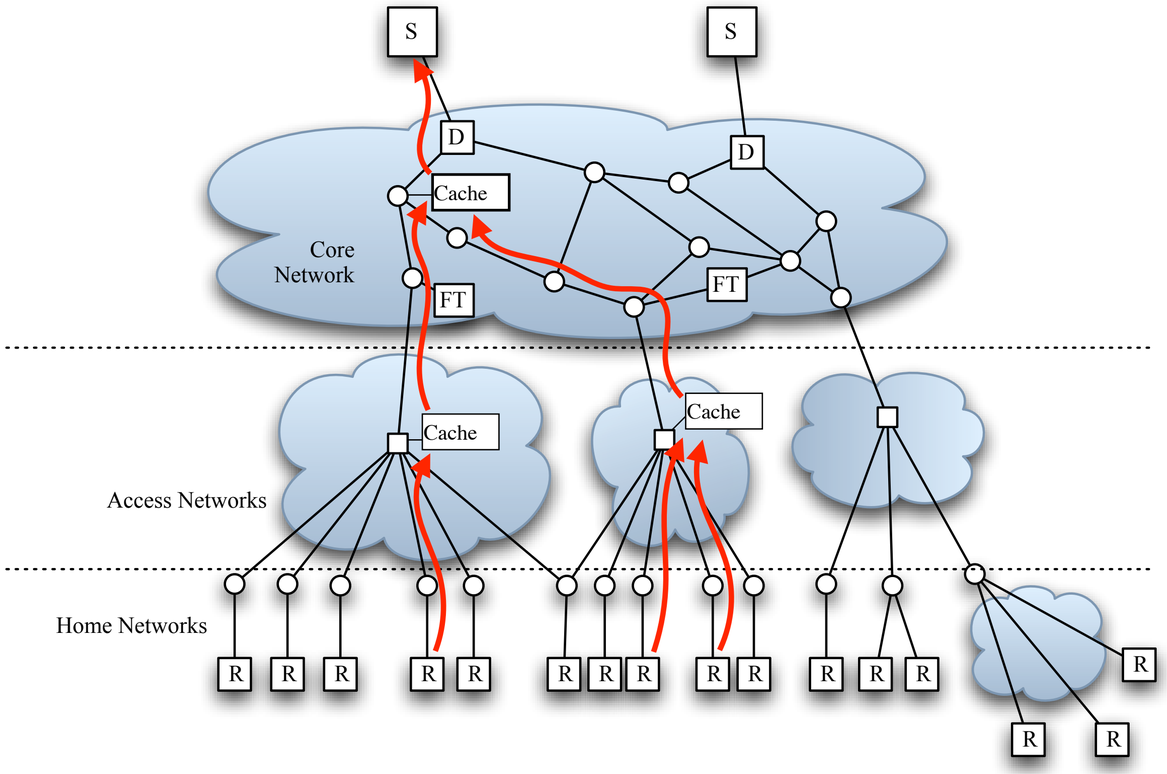
- There is an extensive, well-developed, content distribution infrastructure

Akamai CLOUDFLARE fastly

  - Highly effective for delivering and caching HTTP content
  - Global deployment – agreements with most large ISPs to host caching proxy servers
  - **Only works with HTTP-based content** – does not support RTP-based streaming

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

HTTP Adaptive Streaming (MPEG DASH)

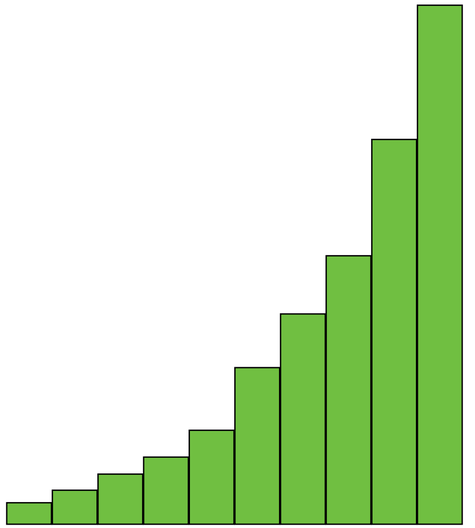


- **Deliver video over HTTPS** to make use of content distribution networks
  - Video content encoded in multiple **chunks**
    - Each chunk encodes ~10 seconds of video
    - Each chunk independently decodable
    - Each chunk made available in many different versions at different encoding rates
  - A **manifest** file provides an index for the chunks that are available
  - Clients fetch manifest, download chunks in turn
    - Standard HTTPS downloads
    - But monitor download rate, and choose what encoding rate to fetch next – adaptive bitrate
    - Playout chunks in turn, as they download

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

DASH Rate Adaptation

- Each chunk encodes ~10 seconds of video content
- Each chunk is compressed multiple times at different encoding rates (i.e., different sizes)
  - e.g., Netflix suggests encoding at ten different rates,
- Receiver fetches manifest, containing the list of all versions of all each chunk
- Receiver fetches each chunk in turn
  - Measure download rate, compare to encoding rate
  - If download rate slower than encoding rate, switch to lower encoding rate for next chunk
  - If download rate faster than encoding rate, consider fetching higher rate for next chunk
- Chooses encoding rate to fetch based on TCP download rate



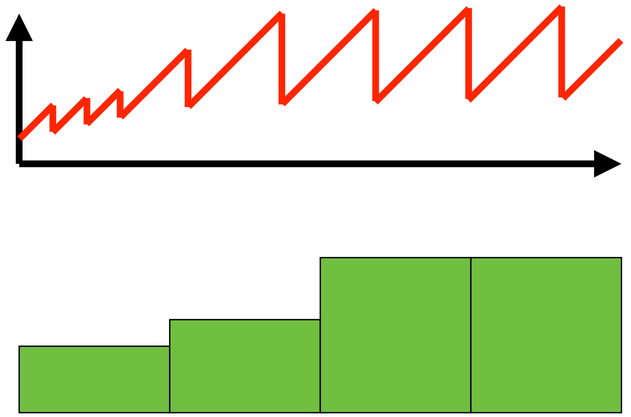
Bitrate (kbps)	Resolution
235	320x240
375	384x288
560	512x384
750	512x384
1050	640x480
1750	720x480
2350	1280x720
3000	1280x720
4300	1920x1080
5800	1920x1080

Source: Netflix

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

DASH and TCP Congestion Control

- DASH rate adaptation and TCP congestion control work at different time scales
  - TCP adjusts congestion window once per RTT
  - DASH receiver measures **average download speed** of TCP connection over ~10 seconds
    - Selects size of next chunk to download based on average TCP download speed



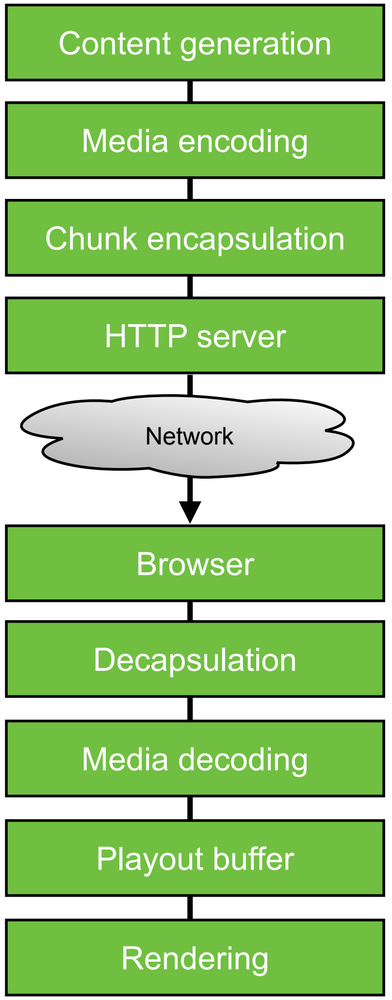
- Videos played using DASH often have relatively poor quality for first few seconds
  - Receiver picked a conservative download rate – relatively poor quality, small size – to start
  - Uses that to measure connection download speed, then switches to more appropriate rate

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

DASH Latency

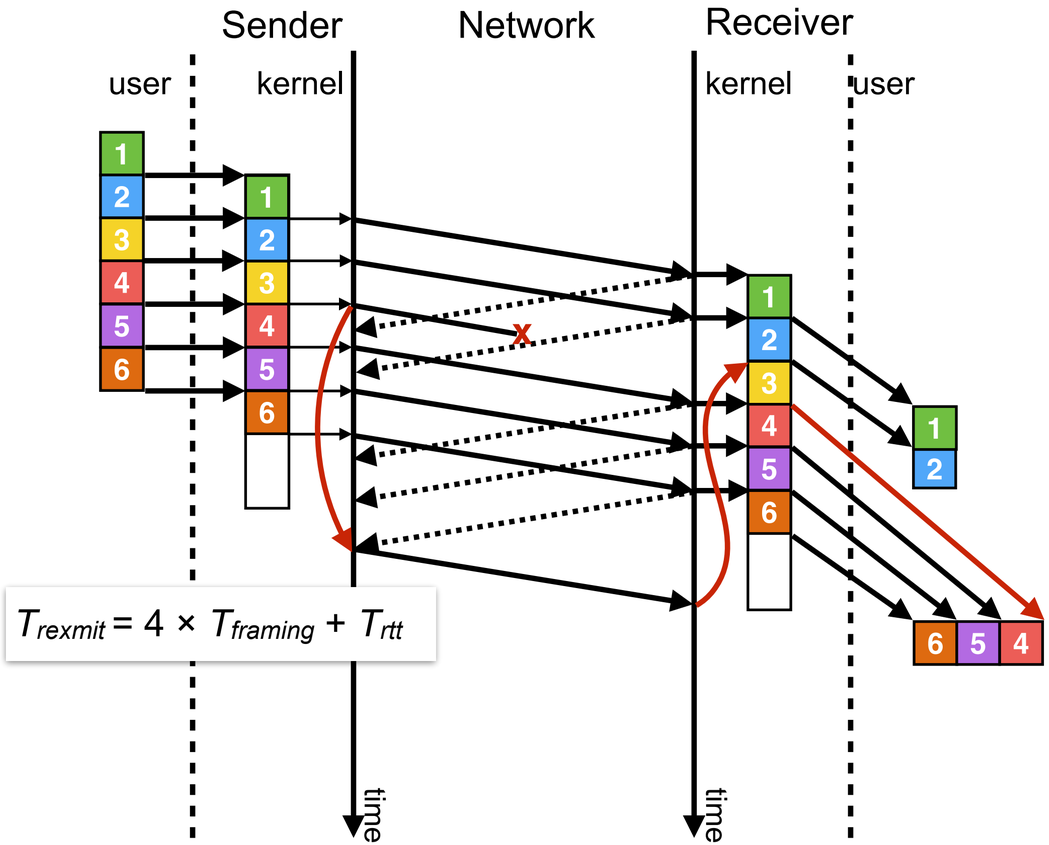
- Multi-second playout latency common, due to:
  - Chunk duration
  - Playout buffering
  - Encoding delays for live streaming

- **Chunk duration is a key** – must download complete chunk before starting playout
  - Download and decompress chunk *n*
  - Start playing chunk *n* and while playing download chunk *n+1*
  - No packets lost → latency equals chunk duration



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Sources of Latency: TCP

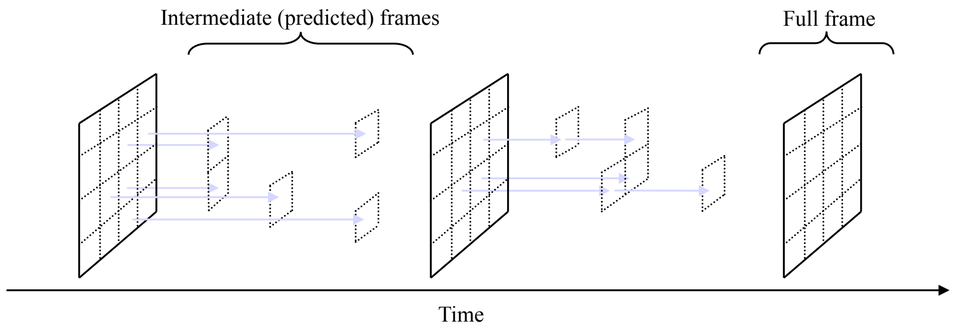


- If a packet is lost, TCP retransmits after triple duplicate ACK
  - This takes time, and delays download of the chunk:
    - 4× packet transmission latency (the lost packet, plus three following that generate duplicate ACKs)
    - 1× network RTT to retransmit packet
  - Each packet loss and retransmission will increase chunk download time – and causes TCP to reduce its window
- Both add latency if using TCP

Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

Sources of Latency: Compression

- Each chunk of video is independently decodable
  - Can’t compress based on previous chunk, because what chunk was previously downloaded varies with the network capacity
- Each chunk therefore starts with an I-frame, followed by P-frames
  - I-frames are large: often 20× size of P-frames
  - If chunks are smaller duration → contain fewer P-frames compared to number of I-frames; video compression ratio goes down
- Trade-off between chunk size and compression overhead
  - Large chunks require less data, but add latency
  - Small chunks reduce latency, but need more data
  - Overheads become excessive for chunk duration ≲2 seconds



Colin Perkins | https://csperkins.org/ | Copyright © 2026 University of Glasgow

University of Glasgow

Real-time and Interactive Applications

- Real-time traffic and its constraints
- Interactive applications → WebRTC
- Streaming applications → DASH
- Custom infrastructure for low latency; CDN and HTTP if latency less critical

CC BY ND